LAPORAN SEMENTARA

Nama : Nur Amri

NIM : 234308106

Kelas : TKA-6D

A. Pendahuluan

MediaPipe merupakan framework yang dikembangkan oleh Google untuk memudahkan pembangunan alur pemrosesan data persepsi dari berbagai sumber, terutama audio dan video. Framework ini dirancang agar pengembang dapat menambahkan kemampuan kecerdasan buatan ke dalam aplikasi tanpa harus merancang seluruh sistem pemrosesan dari awal. Google telah memanfaatkan MediaPipe untuk kebutuhan internal sejak sekitar tahun 2012 dan kemudian merilisnya sebagai proyek open source pada tahun 2019 sehingga dapat digunakan secara luas. Di dalamnya tersedia berbagai solusi berbasis machine learning, seperti deteksi wajah, pelacakan iris, segmentasi rambut, serta analisis pose dan gerakan tubuh secara menyeluruh. Setiap solusi dikemas dalam bentuk modul yang siap diintegrasikan sehingga mempercepat proses pengembangan aplikasi visi komputer. MediaPipe juga mendukung berbagai platform seperti Android dan iOS serta kompatibel dengan beberapa bahasa pemrograman utama, di antaranya C++, Python, dan JavaScript, termasuk dukungan untuk perangkat akselerator seperti Coral. Pada praktikum ini, MediaPipe diterapkan menggunakan bahasa pemrograman Python yang dipadukan dengan pustaka OpenCV untuk melakukan deteksi tangan melalui kamera webcam secara real-time. Kegiatan praktikum tersebut difokuskan untuk mempelajari prinsip kerja sistem deteksi berbasis AI, tahapan pengolahan citra digital, serta bagaimana integrasi antara OpenCV dan MediaPipe dapat digunakan untuk membangun aplikasi deteksi sederhana yang berjalan langsung pada aliran video..

B. Tujuan

- Memahami konsep dasar *computer vision*
- Mempelajari cara kerja pipeline pemrosesan *computer vision* secara *real-time*
- Mengintegrasikan *library* MediaPipe dengan OpenCV dalam aplikasi *Python*
- Mengimplementasikan penggunaan MediaPipe untuk deteksi objek, khususnya deteksi tangan
- Memahami alur akuisisi citra, preprocessing, hingga ekstraksi fitur visual

C. Manfaat

- Menambah keterampilan praktis khususnya di bidang *computer vision* dan AI terapan
- Meningkatkan pemahaman integrasi berbagai *library* pemrosesan citra dalam satu aplikasi
- Memperkuat kemampuan analisis dan pemecahan masalah dalam pengolahan data visual
- Menjadi dasar untuk pengembangan proyek lanjutan seperti *gesture recognition*, *human–computer interaction*, dan sistem monitoring visual

D. Kode Program dan Analisis

1. Deteksi Tangan

Bagian 1 — Inisialisasi pustaka dan perangkat (setup awal program)

```
import sys
print("Interpreter:", sys.executable)

import cv2
import mediapipe as mp

# Inisialisasi kamera
cap = cv2.VideoCapture(0)

# Inisialisasi mediapipe (tanpa drawing landmark)
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7
)
```

Bagian awal program menyiapkan lingkungan kerja dan seluruh pustaka yang diperlukan. Modul sistem dipanggil untuk menampilkan interpreter Python yang sedang digunakan agar dapat dipastikan environment sudah benar. Selanjutnya OpenCV dimuat untuk menangani pengambilan dan penampilan citra, sedangkan MediaPipe digunakan sebagai mesin deteksi tangan berbasis AI. Kamera diaktifkan sebagai sumber video real-time. Modul MediaPipe Hands kemudian diinisialisasi dengan pembatasan satu tangan dan ambang kepercayaan deteksi tertentu agar proses lebih stabil. Pada versi ini tidak diaktifkan utilitas drawing landmark karena program hanya berfokus pada status deteksi, bukan visualisasi titik tangan.

Bagian 2 — Akuisisi frame dan konversi format citra (pra-pemrosesan)

```
while True:
    success, img = cap.read()
    if not success:
        break
```

```
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)
```

Bagian ini menjalankan perulangan utama untuk membaca frame dari webcam secara terus-menerus. Setiap iterasi mengambil satu frame dan memeriksa apakah proses pembacaan berhasil. Jika gagal, perulangan dihentikan untuk menghindari error lanjutan. Frame yang diperoleh masih dalam format warna BGR khas OpenCV, sehingga dikonversi ke format RGB agar sesuai dengan kebutuhan model MediaPipe. Setelah konversi, frame tersebut diproses oleh modul deteksi tangan untuk memperoleh hasil analisis.

Bagian 3 — Logika deteksi tangan dan penentuan status

```
    # HANYA cek ada tangan atau tidak
    if results.multi_hand_landmarks:
        status = "Tangan Terdeteksi"
        print(status)
        warna = (0, 255, 0)
    else:
        status = "Tidak Ada Tangan"
        print(status)
        warna = (0, 0, 255)
```

Bagian ini berisi logika utama pengambilan keputusan berdasarkan hasil pemrosesan MediaPipe. Program memeriksa apakah terdapat data landmark tangan pada output model. Jika ada, sistem menyimpulkan bahwa tangan terdeteksi lalu menetapkan teks status dan warna hijau sebagai indikator visual. Jika tidak ada landmark yang ditemukan, status diubah menjadi tidak ada tangan dengan indikator warna merah. Informasi status juga dicetak ke terminal sebagai log sehingga proses deteksi dapat dipantau. Pada tahap ini tidak dilakukan penggambaran titik landmark, sehingga komputasi lebih ringan dan fokus hanya pada keberadaan objek tangan.

Bagian 4 — Penampilan hasil dan terminasi program

```
  cv2.putText(img, status, (10, 40),
                cv2.FONT_HERSHEY_SIMPLEX,
                1,
                warna,
                2)

    cv2.imshow("Deteksi Tangan Tanpa Landmark", img)

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Bagian akhir bertugas menampilkan hasil deteksi ke layar dan mengelola penghentian program. Status deteksi dituliskan pada frame menggunakan fungsi penambahan teks dengan pengaturan posisi, ukuran huruf, warna, dan ketebalan garis. Frame kemudian ditampilkan pada jendela video sehingga pengguna dapat melihat hasil secara real-time. Program juga memantau input keyboard dan menyediakan perintah keluar cepat saat tombol tertentu ditekan. Setelah perulangan dihentikan, kamera dilepas dan seluruh jendela tampilan ditutup agar sumber daya perangkat tidak tetap terkunci oleh aplikasi.

2. Deteksi Landmark Tangan

Bagian 1 — Inisialisasi pustaka dan perangkat (setup awal program)

```
import sys
print("Interpreter:", sys.executable)

import cv2
import mediapipe as mp

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
hands = mp_hands.Hands(max_num_hands=1)
mp_draw = mp.solutions.drawing_utils
```

Bagian ini menyiapkan seluruh komponen yang dibutuhkan sebelum proses deteksi dijalankan. Program memanggil modul sistem untuk menampilkan interpreter Python yang aktif sehingga dapat dipastikan environment yang digunakan sudah sesuai. Selanjutnya diimpor pustaka OpenCV untuk pengolahan citra dan MediaPipe untuk deteksi tangan berbasis AI. Kamera diaktifkan sebagai sumber video real-time. Setelah itu modul MediaPipe Hands diinisialisasi untuk membuat objek pendeteksi tangan dengan batas satu tangan, serta utilitas drawing disiapkan untuk menggambar landmark pada frame hasil deteksi.

Bagian 2 — Akuisisi frame dan konversi format citra (pra-pemrosesan)

```
while True:
    success, img = cap.read()
    if not success:
        break

    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)
```

Pada bagian ini program menjalankan perulangan utama untuk membaca frame dari webcam secara terus-menerus. Setiap iterasi mengambil satu gambar dan memeriksa keberhasilannya. Jika frame tidak tersedia, perulangan dihentikan. Gambar yang diperoleh masih dalam format warna BGR sehingga dikonversi ke RGB agar sesuai

dengan format masukan yang dibutuhkan oleh model MediaPipe. Frame yang sudah dikonversi kemudian dikirim ke modul deteksi tangan untuk dianalisis.

Bagian 3 — Deteksi tangan dan pemrosesan landmark (inti analisis)

```
if results.multi_hand_landmarks:
    print("Landmark terdeteksi")

    for hand_landmarks in results.multi_hand_landmarks:
        mp_draw.draw_landmarks(
            img,
            hand_landmarks,
            mp_hands.HAND_CONNECTIONS
        )

        for idx, titik in enumerate(hand_landmarks.landmark):
            print("ID:", idx, "X:", titik.x, "Y:", titik.y)
```

Bagian ini merupakan inti proses deteksi. Program memeriksa apakah MediaPipe menemukan landmark tangan pada frame. Jika terdeteksi, sistem menampilkan pesan ke terminal lalu menggambar titik-titik landmark beserta garis penghubung antar sendi tangan pada gambar. Setiap titik landmark kemudian diiterasi untuk diambil nomor indeks dan koordinat posisinya, lalu dicetak. Koordinat yang dihasilkan bersifat ternormalisasi terhadap ukuran frame, sehingga dapat digunakan untuk analisis lanjutan seperti pengenalan gesture. Jika tidak ada tangan yang terdeteksi, program hanya menampilkan pesan bahwa landmark tidak ditemukan.

Bagian 4 — Visualisasi hasil dan terminasi program (output dan penutup)

```
else:
        print("Tidak ada landmark")

    cv2.imshow("Landmark Tangan", img)

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Bagian terakhir bertanggung jawab menampilkan hasil pemrosesan ke jendela tampilan sehingga pengguna dapat melihat landmark tangan yang tergambar secara real-time. Program juga membaca input keyboard dalam selang waktu singkat dan menyediakan tombol keluar cepat dengan menekan huruf tertentu agar proses dapat dihentikan secara manual. Setelah perulangan selesai, kamera dilepas dan seluruh jendela tampilan ditutup untuk memastikan tidak ada sumber daya perangkat yang masih digunakan oleh aplikasi.

3. Deteksi Tangan Kanan/Kiri

Bagian 1 — Inisialisasi pustaka, kamera, dan modul deteksi

```
import sys
print("Interpreter:", sys.executable)

import cv2
import mediapipe as mp

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
hands = mp_hands.Hands(max_num_hands=1)
mp_draw = mp.solutions.drawing_utils
```

Bagian awal ini menyiapkan seluruh kebutuhan program sebelum proses deteksi dijalankan. Interpreter Python ditampilkan untuk memastikan environment yang aktif sudah benar. Selanjutnya dimuat pustaka OpenCV untuk pengolahan video dan MediaPipe untuk deteksi tangan berbasis AI. Kamera diaktifkan sebagai sumber input visual secara real-time. Modul MediaPipe Hands diinisialisasi untuk mendeteksi maksimal satu tangan agar pemrosesan lebih ringan. Selain itu, drawing utilities disiapkan untuk menggambar titik landmark dan garis koneksi tangan di atas frame hasil tangkapan kamera.

Bagian 2 — Perulangan pembacaan frame dan pra-pemrosesan citra

```
while True:
    success, img = cap.read()
    if not success:
        break

    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)

    if results.multi_hand_landmarks:
```

Pada bagian ini program menjalankan loop utama untuk membaca frame video secara terus-menerus dari webcam. Setiap frame diperiksa keberhasilannya agar hanya data valid yang diproses. Gambar yang diperoleh masih dalam format BGR sehingga dikonversi terlebih dahulu menjadi RGB supaya sesuai dengan format masukan model MediaPipe. Frame RGB tersebut kemudian diproses oleh modul deteksi tangan untuk menghasilkan data landmark dan klasifikasi sisi tangan.

Bagian 3 — Deteksi landmark, klasifikasi kanan/kiri, dan pembalikan label

```
for idx, hand_landmarks in enumerate(results.multi_hand_landmarks):

        mp_draw.draw_landmarks(
```

```
                img,
                hand_landmarks,
                mp_hands.HAND_CONNECTIONS
            )

            label =
    results.multi_handedness[idx].classification[0].label

            if  label == "Right":
                label = "Left"
            else:
                label = "Right"

            print("Tangan:", label)

            cv2.putText(img, f"Tangan: {label}", (10,40),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 2)

    else:
        print("Tidak ada tangan")
```

Bagian ini merupakan inti analisis deteksi. Program memeriksa apakah landmark tangan ditemukan. Jika ada, setiap tangan yang terdeteksi diproses dengan indeksnya agar data landmark dan data klasifikasi bisa dipasangkan dengan benar. Sistem kemudian menggambar titik-titik landmark dan garis hubungan antar sendi tangan pada frame. Setelah itu program mengambil label klasifikasi tangan dari MediaPipe, yaitu kanan atau kiri. Karena tampilan kamera biasanya bersifat mirror, label tersebut dibalik agar sesuai dengan persepsi pengguna di layar. Label akhir ditampilkan di terminal dan juga dituliskan pada frame video sebagai informasi visual. Jika tidak ada tangan yang terdeteksi, program hanya menampilkan pesan ketiadaan tangan.

Bagian 4 — Penampilan hasil dan penutupan program

```
        cv2.imshow("Deteksi Kanan/Kiri", img)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Bagian terakhir menangani tampilan output dan terminasi aplikasi. Frame yang telah diberi gambar landmark dan label sisi tangan ditampilkan pada jendela video sehingga hasil deteksi dapat diamati secara real-time. Program memeriksa input keyboard secara berkala dan menyediakan perintah keluar cepat saat tombol tertentu ditekan. Setelah loop dihentikan, koneksi kamera dilepaskan dan seluruh jendela tampilan ditutup

untuk memastikan sumber daya perangkat keras dan antarmuka grafis dibersihkan dengan benar.

4. Deteksi Tangan Depan/Belakang

Bagian 1 — Inisialisasi pustaka, kamera, dan modul deteksi

```
import sys
print("Interpreter:", sys.executable)

import cv2
import mediapipe as mp

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
hands = mp_hands.Hands(max_num_hands=1)
mp_draw = mp.solutions.drawing_utils
```

Bagian pertama menyiapkan seluruh dependensi dan perangkat yang diperlukan program. Interpreter Python ditampilkan untuk memastikan lingkungan eksekusi sudah sesuai. Pustaka OpenCV dimuat untuk menangani akuisisi dan tampilan video, sedangkan MediaPipe digunakan sebagai mesin deteksi tangan berbasis model AI. Kamera diaktifkan sebagai sumber data visual real-time. Modul MediaPipe Hands diinisialisasi untuk mendeteksi satu tangan, dan drawing utilities disiapkan agar landmark tangan serta garis koneksinya dapat digambar di atas frame hasil tangkapan kamera.

Bagian 2 — Akuisisi frame, pembalikan citra, dan konversi warna

```
while True:
    success, img = cap.read()
    if not success:
        break
    image = cv2.flip(img, 0)
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)
```

Bagian ini menjalankan perulangan utama untuk membaca frame dari webcam secara kontinu. Setiap frame diperiksa keberhasilannya sebelum diproses lebih lanjut. Terdapat operasi pembalikan citra menggunakan fungsi flip yang membalik orientasi frame terhadap sumbu tertentu, yang umumnya dipakai untuk menyesuaikan sudut pandang tampilan kamera. Setelah itu frame dikonversi dari format warna BGR menjadi RGB agar sesuai dengan format input model MediaPipe. Frame RGB kemudian dikirim ke modul deteksi untuk dianalisis dan diambil data landmark tangannya.

Bagian 3 — Deteksi landmark dan penentuan empat kondisi tangan

```
if results.multi_hand_landmarks:

    for idx, hand_landmarks in
```

```
enumerate(results.multi_hand_landmarks):

        mp_draw.draw_landmarks(
            img,
            hand_landmarks,
            mp_hands.HAND_CONNECTIONS
        )

        handedness =
results.multi_handedness[idx].classification[0].label

        thumb_tip = hand_landmarks.landmark[4]
        pinky_tip = hand_landmarks.landmark[20]


        if handedness == "Right":

            if thumb_tip.x < pinky_tip.x:
                kondisi = "Tangan Kiri - Telapak (Depan)"
            else:
                kondisi = "Tangan Kiri - Punggung (Belakang)"

        else:  # Left

            if thumb_tip.x > pinky_tip.x:
                kondisi = "Tangan Kanan - Telapak (Depan)"
            else:
                kondisi = "Tangan Kanan - Punggung (Belakang)"

        print("Kondisi:", kondisi)

        cv2.putText(img,
                    kondisi,
                    (10, 40),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    0.8,
                    (0, 255, 0),
                    2)
```

Bagian ini merupakan inti analisis deteksi dan klasifikasi kondisi tangan. Program memeriksa apakah landmark tangan terdeteksi, lalu memproses setiap tangan yang ditemukan. Landmark digambar di atas frame untuk memperlihatkan struktur titik dan koneksi jari. Sistem mengambil informasi sisi tangan (kanan/kiri) dari hasil klasifikasi MediaPipe, kemudian mengekstrak koordinat ujung ibu jari dan ujung kelingking. Perbandingan posisi horizontal kedua titik tersebut digunakan sebagai dasar logika untuk menentukan apakah yang terlihat adalah telapak atau punggung tangan. Karena tampilan kamera bersifat mirror, logika label kanan–kiri dibalik agar sesuai dengan persepsi visual pengguna. Hasil klasifikasi kondisi ditampilkan di terminal dan juga dituliskan pada frame sebagai teks.

Bagian 4 — Penanganan tanpa deteksi, tampilan output, dan terminasi

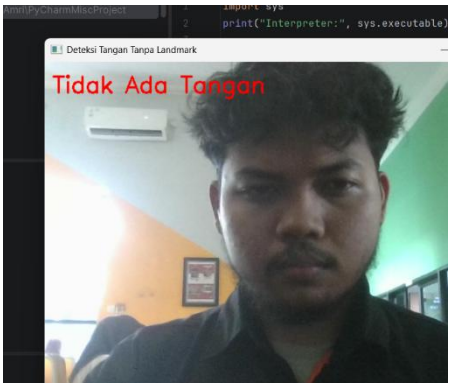
```
    else:
          cv2.putText(img,
                      "Tidak Ada Tangan",
                      (10, 40),
```

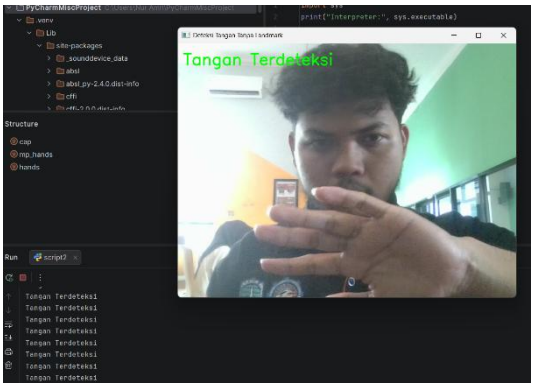
```
                                cv2.FONT_HERSHEY_SIMPLEX,
                                0.8,
                                (0, 0, 255),
                                2)

            cv2.imshow("Deteksi 4 Kondisi Tangan", img)

            if cv2.waitKey(10) & 0xFF == ord('q'):
                break

    cap.release()
    cv2.destroyAllWindows()
```

Bagian akhir mengatur respons ketika tidak ada tangan yang terdeteksi dengan menampilkan pesan khusus pada frame video. Selanjutnya frame hasil pemrosesan ditampilkan pada jendela output sehingga pengguna dapat memantau deteksi empat kondisi tangan secara real-time. Program juga membaca input keyboard secara berkala dan menyediakan tombol keluar cepat untuk menghentikan proses. Setelah perulangan selesai, kamera dilepaskan dan seluruh jendela ditutup agar sumber daya perangkat tidak tetap terkunci oleh aplikasi.

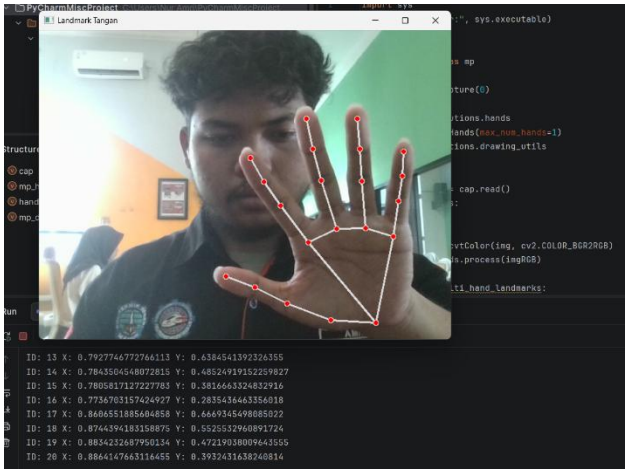
E. Gambar Hasil Pengujian

1. Deteksi Tangan

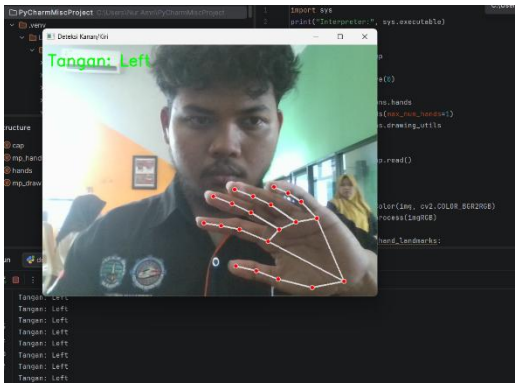


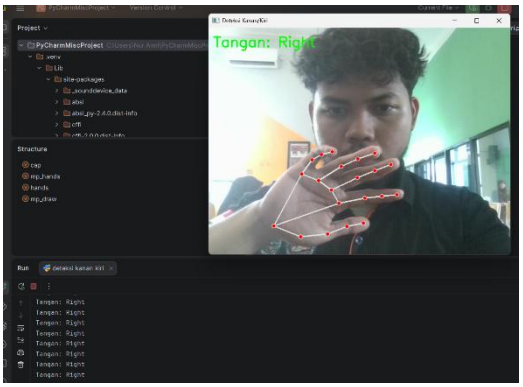


2. Deteksi Landmark

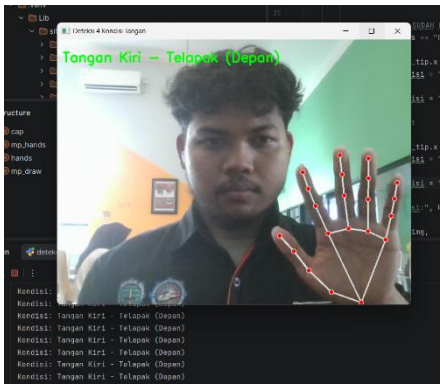


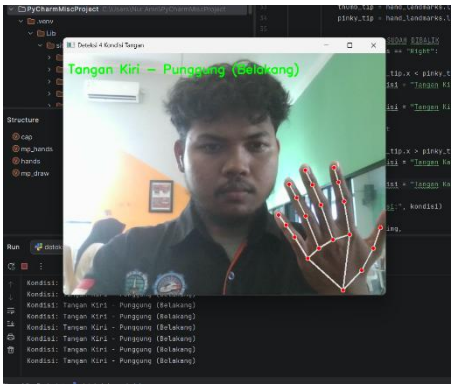
3. Deteksi Tangan Kanan/Kiri

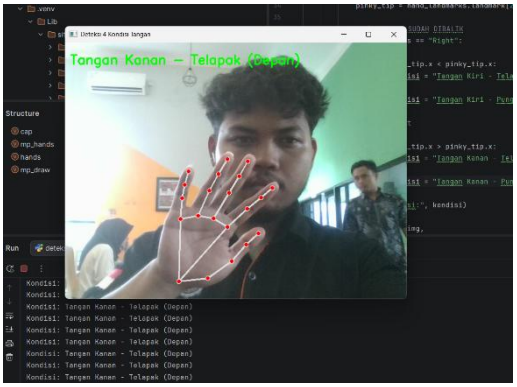


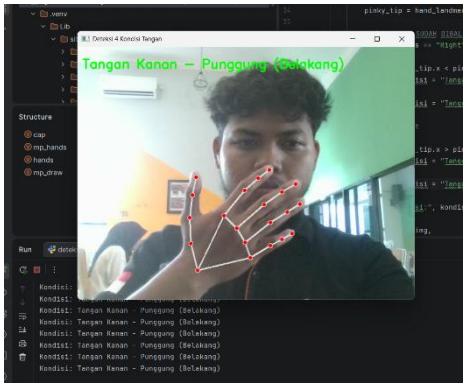


4. Deteksi Tangan Kanan/Kiri dan Depan/Belakang









F. Kesimpulan

- Praktikum ini menunjukkan bahwa framework MediaPipe dapat diintegrasikan dengan OpenCV dan Python untuk membangun sistem deteksi tangan berbasis visi komputer secara real-time melalui webcam.
- Proses yang dilakukan mencakup akuisisi citra dari kamera, konversi format warna, pemrosesan menggunakan model AI, hingga penampilan hasil deteksi pada layar.
- Sistem berhasil mendeteksi keberadaan tangan tanpa landmark maupun dengan visualisasi landmark sehingga struktur titik sendi dan jari dapat diamati.
- Data landmark yang dihasilkan dapat dimanfaatkan untuk analisis lanjutan karena memuat koordinat tiap titik tangan secara terukur.

- Program mampu membedakan sisi tangan kanan dan kiri dengan memanfaatkan fitur klasifikasi bawaan MediaPipe serta penyesuaian logika terhadap efek mirror kamera.
- Pengolahan koordinat ujung ibu jari dan kelingking memungkinkan identifikasi orientasi tangan, sehingga telapak dan punggung tangan dapat dibedakan.
- Beberapa variasi program yang diuji membuktikan bahwa perubahan logika dan parameter dapat memperluas fungsi deteksi tanpa perlu mengganti model dasar.
- Implementasi ini memperlihatkan alur lengkap pipeline computer vision mulai dari input video, preprocessing, inferensi model, sampai visualisasi output.
- Praktikum memberikan pemahaman praktis tentang cara kerja deteksi objek berbasis AI dan integrasi multi-library dalam satu aplikasi.
- Hasil percobaan dapat dijadikan dasar untuk pengembangan lanjutan seperti gesture recognition, human–computer interaction, dan sistem kontrol berbasis gerakan tangan.

G. Referensi

- https://belajarpython.com/
- https://youtu.be/a99p_fAr6e4?si=FTYbwWDeqbjJVSWF
- https://youtu.be/wpNdYjBKvzU?si=pjfvsayEH3BdU2UC